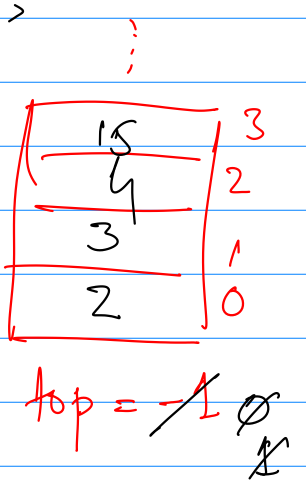
```
class Stack
{   private:
            int  * arr;
        ->  int  top;     //index
            int  size;
    public:
            Stack( int  s)
          {     size = s;
                arr = new  int[size];
                 top = -1;
          }
          ~Stack( )
           {
                delete[] arr;
           }
           void  push(int  val)
           {
              if ( top == (size - 1))   // full
              {
                   cout<<"Overflow";
                   return;
              }
                arr[++top] = val;
                //arr[top++] = val;
           }
```

5
0 — 4

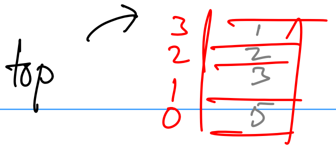
4
0 — 3 push(0)

push(2)
push(3)
push(4)
push(15)

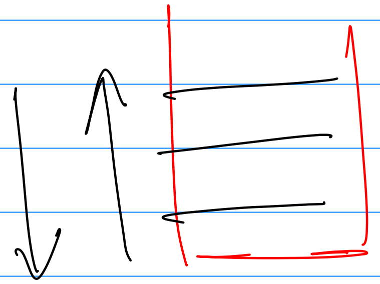


2
3

```
int   pop( )
{
      if( top == -1)
      {
            Cout<<"Underflow";
            return;
      }
      //return   arr[top--];
        int   val = arr[top];
        top--;
        return val;
}
  int  peek()
  {
     //isEmpty {
     retur    arr[top];
  }
};
```



```
class Stack
{
   private:
        linkedlist        l;
int     type(?)  top;
```



l.insert(val, pos);

int

```
Class   Stack
{
   private:
           linkedlist  l;

   public:
        void   push( int  val)
        {
           l.insert (val , 1); ✓
        }
        int   pop ( )
        {
           if (l.is Empty())
           {
                Cout<< " Underflow";
                return  -105;
           }
            int  val  = l.get(1);
            l. remove (1);
            return  val;
        }

   };
```